FAST COMPUTATION OF OVERFLOW FLAG IN A BIT MANIPULATION UNIT

ABSTRACT OF THE DISCLOSURE

A bit manipulation unit (BMU) scales and formats data and includes fast computation of the overflow flag. For fast computation the BMU's overflow flag is computed based on the input data and the shift amount. The overflow flag is calculated separately as either a LMV_{left} for an arithmetic shift left operation or LMV_{right} for an arithmetic shift right operation. For an arithmetic shift left operation, LMV_{left} may be computed by first adding one plus the number of guard bits in the input data to the shift amount, and then detecting the number of redundant sign bits. For an arithmetic shift right operation, LMV_{right} may be computed by checking the input redundant sign bits plus the right shift amount. By computing the overflow flag separately as LMV_{left} and LMV_{right} for arithmetic left and right shifts, respectively, the overflow flag LMV is determined in parallel with the barrel shift operation and so does not depend on the result from the barrel shift operation. Consequently, an advantage of employing this technique in a BMU may be a relative reduction in the time necessary for a BMU to calculate the overflow flag.